Marquette Intellectual Property Law Review

Volume 10 | Issue 3 Article 4

Reverse Engineering of Computer Programs under the DMCA: Recognizing a "Fair Access" Defense

Donna L. Lee

WINNER OF THE COMPUTER LAW ASSOCIATION 2005 INFORMATION TECHNOLOGY LAW WRITING COMPETITION

Reverse Engineering of Computer Programs Under the DMCA: Recognizing a "Fair Access" Defense

INTRODUCTION

Reverse engineering of computer programs is a widely-practiced, industry-accepted way of achieving several different objectives. Often the purpose of reverse engineering is to develop an interoperable program. Other purposes include customizing a program for the user's needs, fixing bugs, detecting infringement, or simply studying a program. These purposes might be mixed. For example, a company will frequently reverse engineer a competitor's new program; first, to see if it infringes on any of the company's own programs, but beyond this, to observe, for example, how the competitor dealt with a constraint imposed by industry standards for similar programs. Even if the company never incorporates these observations into one of its own programs, the act of reverse engineering itself spurs innovation. Like music, computer programming is a performing art (the performers are machines), and, like composers, programmers analyze how programs written by other people perform in order to hone their own creative skills.

Despite its widespread application, there is currently some uncertainty as to the lawfulness of reverse engineering in the United States.[1] Computer programs are copyrighted works, and reverse engineering necessarily entails making copies of an entire program, thereby infringing on the reproduction right of the copyright owner.[2] Courts have consistently held, though, that when a defendant has a legitimate purpose and no other means of accessing the unprotected elements of a program, reverse engineering constitutes fair use.[3] Under

1. Elsewhere, the lawfulness of reverse engineering computer programs has also been called into question. *See* Céline M. Guillou, *The Reverse Engineering of Computer Software in Europe and the United States: A Comparative Approach*, 22 COLUM.-VLA J.L. & ARTS 533 (1998); Aashit Shah, *UK's Implementation of the Anti-Circumvention Provisions of the EU Copyright Directive: An Analysis*, 2004 DUKE L. & TECH. REV. 3, *at* http://www.law.duke.edu/journals/dltr/articles/2004dltr0003.html.

2. 17 U.S.C. § 106(1) (2000). Reverse engineering might also run up against contract and licensing claims, because a program is often transferred under an agreement that explicitly forbids the licensee/purchaser to reverse engineer the program. *See* Lydia Pallas Loren, *Slaying the Leather-Winged Demons in the Night: Reforming Copyright Owner Contracting with Clickwrap Misuse*, 30 OHIO N.U. L. REV. 495, 508–12 (2004) (reviewing the doctrines of unconscionability and preemption as they apply to clickwrap licenses). This Comment will not address contract and licensing claims, but will focus solely on claims brought under the Copyright Act and the DMCA.

3. 17 U.S.C. § 107 (2000); *see* Sega Enters. Ltd. v. Accolade, Inc., 977 F.2d 1510, 1518 (9th Cir. 1992).

> [W]e conclude based on the policies underlying the Copyright Act that disassembly

the Digital Millennium Copyright Act (DMCA), which prohibits the circumvention of "a technological measure that effectively controls access to a [copyrighted] work,"[4] reverse engineering does not fair as well. Computer programs are considered "technology" within the meaning of the DMCA.[5] Furthermore, "circumvention" is defined in the DMCA as descrambling, decrypting, or otherwise avoiding, bypassing, removing, deactivating, or impairing—in short, reverse engineering—a technological measure without the authority of the copyright owner.[6] The DMCA does provide an exemption for reverse engineering "for the sole purpose of . . . interoperability."[7] But the exemption is beset with ambiguities and too narrowly crafted to accommodate the many different purposes of reverse engineering.[8]

This Comment argues that, in light of Congress's express intent to codify the settled law regarding reverse engineering under the Copyright Act, a court should resolve the ambiguities in the DMCA's reverse engineering exemption in favor of the defendant. In addition, courts should develop a "fair access" defense for reverse engineering undertaken for purposes that do not involve interoperability, but rather enable other reasonable, fair-use-defensible uses of computer programs.

Part I of this Comment briefly describes the practice of reverse engineering. Part II summarizes how courts have dealt with reverse engineering under the Copyright Act. Part III discusses how courts should deal with reverse engineering under the DMCA. To begin, Part III considers the legislative history of the DMCA. Next, Part III looks closely at the reverse engineering exemption that Congress provided in § 1201(f) of the DMCA. This Comment discusses three ambiguities a court must resolve in order to apply § 1201(f): (1) the scope of the reverse engineer's task (Scope), (2) whether a reverse engineer may

> of copyrighted object code is, as a matter of law, a fair use of the copyrighted work if such disassembly provides the only means of access to those elements of the code that are not protected by copyright and the copier has a legitimate reason for seeking such access.

Id.

4. 17 U.S.C. § 1201(a)(1)(A) (2000).

5. *See* Universal City Studios, Inc. v. Reimerdes, 111 F. Supp. 2d 294, 317 (S.D.N.Y. 2000), *aff'd*, Universal City Studios, Inc. v. Corley, 273 F.3d 429 (2d Cir. 2001) ("DeCSS, a computer program, unquestionably is 'technology' within the meaning of the [DMCA].").

6. § 1201(a)(3)(A).

7. § 1201(f)(1).

8. *See, e.g.*, Dan L. Burk, *Anticircumvention Misuse*, 50 UCLA L. REV. 1095, 1106 (2003) (commenting that "the explicit exceptions to the circumvention provisions have been correctly criticized as narrow and shortsighted, failing to anticipate any new or unexpected reason that users might legitimately have for needing access to a work").

embed a circumvention tool in a new, interoperable program without violating the "anti-trafficking" provisions of the DMCA (Tools),[9] and (3) whether a court must establish the noninfringing character of a new, interoperable program before it applies the reverse engineering exemption (Character). Finally, Part III argues that the DMCA leaves room for courts to develop a "fair access" defense for reverse engineering that is justifiable for reasons that do not involve interoperability. The Comment discusses how precedent for a fair access defense may be found in two recent appellate court opinions,[10] specifically in the courts' interpretation of the words "access," "protection," and "authority" in § 1201(a). The Comment goes on to suggest three factors courts should weigh when considering the fair access defense: (1) whether the access in question led to what traditionally would be considered a fair use of the program, (2) whether an inherent limitation in the market led to the defendant's need to use self-help to gain access, and (3) whether the nature of the plaintiff's program is such that it deserves only relatively weak protection under the DMCA.

I. REVERSE ENGINEERING OF COMPUTER PROGRAMS

The Supreme Court has defined reverse engineering as "fair and honest means . . . by starting with the known product and working backwards to divine the process which aided in its development or manufacture."[11] In concept, reverse engineering of computer programs is little different from the reverse engineering that takes place in many other contexts.[12] To draw again on the analogy suggested above, a programmer reverse engineering a computer program is like a composer who decides to analyze a piece of music written for a symphony orchestra. Perhaps the composer has been asked to write a piece in a similar style; perhaps she simply likes the piece and wants to understand why it moves her. The composer can (1) read what others have written about the music, (2) listen to a recording of the music, (3) transcribe the

9. § 1201(a)(2), (b)(1).

10. Chamberlain Group, Inc. v. Skylink Techs., Inc., 381 F.3d 1178 (Fed. Cir. 2004); Lexmark Int'l, Inc. v. Static Control Components, Inc., 387 F.3d 522 (6th Cir. 2004).

11. Kewanee Oil Co. v. Bicron Corp., 416 U.S. 470, 476 (1974).

12. *See* Andrew Johnson-Laird, *Reverse Engineering of Software: Separating Legal Mythology from Actual Technology*, 5 SOFTWARE L.J. 331, 334–35 (1992) (comparing reverse engineering to recreating a recipe from the taste of the dish, deciding how to repair a car from the sound coming from under the hood, and diagnosing a medical condition from the patient's description of the symptoms).

recorded sounds,[13] and, finally, (4) listen again, going back over the transcription and making other structural diagrams of how the music unfolds. The composer will need to repeat the last two steps several times, working out details of instrumentation, articulation, rhythm, phrase structure, and harmonic progression, to conceptualize as precisely as possible the sounds she hears. In the end, of course, the transcription and diagrams will merely represent what the composer thinks makes the music sound the way it does. But, in the process of making the diagrams, the composer will have formed an idea of what elements of the music are critical to its style, and what is the source of its emotional power.

A programmer who wants to know more about how a computer program operates has four ways to learn about the program, which roughly parallel the four ways a composer can study a piece of music.[14] The programmer can (1) read the product manual or other technical information available on the program, (2) run the program on a computer to observe what it does, (3) use a decompiler to translate all or part of the ones and zeros of the program's machine-readable object code into human-readable words and mathematical symbols known as source code,[15] and, finally, (4) make a "dynamic examination" of the program's code, decompiling parts of the code while the program is running.[16] The process of reverse engineering might be an end unto itself.[17] In addition, the programmer might use the information gathered during steps three and four, for example, to diagnose a bug in the program, develop a new, interoperable program, or use as the basis for an infringement suit against the company that made the program.

Every time a computer program is run, "intermediate" copies of the entire program are made in the computer's random access memory

13. Assume that a copy of the score is not available. Even if the score were available, however, the composer would need to go through the steps outlined here (substituting the score for a transcription) to come to an understanding of how the music was put together.

14. *See* Andrew Johnson-Laird, *Software Reverse Engineering in the Real World*, 19 U. DAYTON L. REV. 843, 846 (1994) (outlining the four ways to reverse engineer computer programs); *see also* Sony Computer Entm't, Inc. v. Connectix Corp., 203 F.3d 596, 599–601 (9th Cir. 2000) (describing the process of reverse engineering).

15. *See* Sega Enters. Ltd. v. Accolade, Inc., 977 F.2d 1510, 1514–15 n.2 (9th Cir. 1992) (describing source code, object code, and decompilers).

16. Johnson-Laird, *supra* note 14, at 846.

17. *See* Johnson-Laird, *supra* note 12, at 346 ("Full time professional programmers probably indulge in reverse engineering at least once a week as part and parcel of their normal job.").

(RAM).[18] Decompiling necessarily entails making multiple intermediate copies. And yet, just as a composer would learn little about how a piece is put together if she were merely to read what others have written about the music and to listen to a recording (steps one and two), so too would a programmer learn little about the inner workings of a computer program if she were merely to read what others have written about the program and to observe what tasks it performs on a computer screen.[19] A programmer must be able to read and perform a dynamic examination of the code (steps three and four)—and must, in the process, be able to make copies of the program—no matter what the purpose of reverse engineering.

Judging from the cases that have been decided, the purpose of reverse engineering is most often to make a new, interoperable program. No doubt this generalization is skewed by the fact that reverse engineering in other contexts does not make its way to court. Broadly speaking, though, programmers reverse engineer for two reasons: to analyze how a program operates and to analyze why a program is not operating properly.[20] Behind both of these reasons lies the fact that much of what goes on when a computer program operates is not normally visible.[21] Furthermore, even what does become visible through reverse engineering presents an incomplete picture of how and why a program operates as it does.[22] Reverse engineering merely

18. *See* Johnson-Laird, *supra* note 14, at 894 (quoting MAI Sys. Corp. v. Peak Computer, Inc., 991 F.2d 511, 519 (9th Cir. 1993)) ("[RAM is] a computer component in which data and computer programs can be temporarily recorded. . . . It is a property of RAM that when the computer is turned off, the copy of the program recorded in memory is lost.").

19. *See id.* at 846–47 (arguing that documentation on computer programs is invariably inadequate and that "the only option guaranteed to provide accurate, complete information is to examine the software itself").

20. *See id.* at 846 (stating that the two reasons to reverse engineer are, first, "to understand how a computer program really works" and, second, "to understand why a program really does *not* work").

21. *See* Sega Enters. Ltd. v. Accolade, Inc., 977 F.2d 1510, 1520 (9th Cir. 1992) ("The need to disassemble object code arises, if at all, only in connection with operations systems, system interface procedures, and other programs that are not visible to the user when operating . . . .").

22. See Johnson-Laird, *supra* note 14, at 896–97.

> Reverse engineering does not lay bare a program's inner secrets. . . . The inner secrets of a program, the real crown jewels, are embodied in the higher levels of abstraction material such as the source code commentary and the specification. This material never survives the process of being converted to object code. As the inner secrets of a program are not in the object code, reverse engineering cannot lay them bare.
>
> . . . In other words, reverse engineering is almost entirely an additive process,

facilitates the analytical, creative thinking programmers must engage in to stay connected to the performance aspects of their art.

II. REVERSE ENGINEERING UNDER THE COPYRIGHT ACT

The status of reverse engineering under the Copyright Act is intimately bound-up with the status of computer programs as protectable under copyright law. As explained above, reverse engineering entails making intermediate copies of the entire computer program. Thus, the reverse engineer necessarily infringes on the reproduction right of the copyright owner of the program.[23] However, not every aspect of a computer program is copyrightable. When the purpose of reverse engineering is to analyze a program in order to study the uncopyrightable aspects of the original, courts have upheld the reverse engineer's right to make intermediate copies under the affirmative defense of fair use.[24]

Courts have grappled with how to distinguish the copyrightable and uncopyrightable aspects of computer programs. Congress expressly extended copyright protection to computer programs in 1980, when it added "computer program" to § 101 of the Copyright Act: "A 'computer program' is a set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result."[25] As this definition suggests, computer programs are protectable

---

with the reverse engineer adding his or her knowledge and experience to the meager information contained within the object code.

Id.

23. 17 U.S.C. § 106(1) (2000). The Copyright Act provides that it is not infringement for someone who owns a computer program to make RAM copies in the normal course of using the program. *Id.* § 117(a)(1). Courts have rejected the argument that § 117 excuses copies made during reverse engineering. *See, e.g.*, *Sega*, 977 F.2d at 1517–18.

24. Fair use is a judicially-created doctrine, codified in § 107 of the 1976 Copyright Act, that provides that a use of a copyrighted work "for purposes such as criticism, comment, new reporting, teaching (including multiple copies for classroom use), scholarship, or research, is not an infringement of copyright." 17 U.S.C. § 107 (2000). Section 107 includes a list of four factors for courts to consider in determining whether a particular use is a fair use:

(1) the purpose and character of the use, including whether such use is of a commercial nature or is for nonprofit educational purposes;

(2) the nature of the copyrighted work;

(3) the amount and substantiality of the portion used in relation to the copyrighted work as a whole; and

(4) the effect of the use upon the potential market for or value of the copyrighted work.

Id.; *see* 4 MELVILLE B. NIMMER & DAVID NIMMER, NIMMER ON COPYRIGHT, § 13.05 (2005) (discussing the fair use doctrine).

25. 17 U.S.C. § 101 (2000). Legislative history reveals that Congress already intended

as "literary works." The lines of code are "statements or instructions," akin to sentences in a book. This analogy is limited, however, because unlike sentences in a book, which are easily distinguished from the unprotected elements of the book, lines of code are not easily distinguished from the unprotected elements of a computer program.[26] A court must somehow separate "idea" from "expression" in the lines of code before it can draw meaningful comparisons between an original and allegedly infringing program.[27]

to include protection for computer programs as literary works when it enacted the 1976 Copyright Act: "The term 'literary works' does not connote any criterion of literary merit or qualitative value . . . . It . . . includes . . . computer programs to the extent that they incorporate authorship in the programmer's expression of original ideas, as distinguished from the ideas themselves." H.R. REP. NO. 94-1476, at 54 (1976). Congress waited to provide express protection for computer programs until it had received word from the Commission on New Technological Uses of Copyrighted Works as to precisely what sort of protection should apply. *See* 5 COPYRIGHT, CONGRESS AND TECHNOLOGY: THE PUBLIC RECORD (Nicholas Henry ed., 1980) (reprinting the Commission's Final Report and Recommendations). In 1980, when it added the definition of computer programs to § 101, Congress also enacted the current version of § 117, "Limitations on exclusive rights: Computer programs." An earlier version of § 117, enacted in 1976, provided that a copyright owner would not receive "any greater or lesser rights" when the copyrighted work was used "in conjunction with automatic systems capable of storing, processing, retrieving, and transferring information." *Id.* at 6.

26. Several authors have criticized the analogy between literary works and computer programs. *See, e.g.*, Julie E. Cohen, *Reverse Engineering and the Rise of Electronic Vigilantism: Intellectual Property Implications of "Lock-Out" Programs*, 68 S. CAL. L. REV. 1091, 1107 (1995) (stating that "the classification of computer programs as 'literary works' is staggeringly uninformative"); Jon O. Newman, *New Lyrics for an Old Melody: The Idea/Expression Dichotomy in the Computer Age*, 17 CARDOZO ARTS & ENT. L.J. 691, 699–700 (1999) ("[A] somewhat different vocabulary might assist us in properly balancing the ultimate values we seek to advance. . . . I would rather begin the process of labeling the protectable and unprotectable elements of computer programs with terms peculiar to the realm of software."); Pamela Samuelson et al., *A Manifesto Concerning the Legal Protection of Computer Programs*, 94 COLUM. L. REV. 2308, 2316–20 (1994) (pointing out that the behavior of a program is more important to consumers than the text, or code, of the program).

27. The rule that copyright law protects expressions and not ideas is codified in § 102(b) of the Copyright Act: "In no case does copyright protection for an original work of authorship extend to any idea, procedure, process, system, method of operation, concept, principle, or discovery, regardless of the form in which it is described, explained, illustrated, or embodied in such work." 17 U.S.C. § 102(b) (2000); *see also* Baker v. Selden, 101 U.S. 99, 101–02 (1879). This rule goes back at least as far as *Baker*, in which the Supreme Court held that the author of a book on a particular system of book-keeping could claim protection for his explanation of the system, but not for the system itself. 101 U.S. at 101–02.

Two doctrines that courts have developed to deal with the idea-expression dichotomy are the doctrines of merger and *scenes a faire*. Under the merger doctrine, an idea that can be expressed in only one or a limited number of ways is said to have "merged" with its expression, so that the expression is either uncopyrightable or receives only "thin" protection. *See* 4 NIMMER & NIMMER, *supra* note 24, § 13.03[B][3] (discussing the merger doctrine).

In one of the earliest cases to deal with the idea/expression dichotomy in computer programs, the Third Circuit concluded that the idea of a program is its overall purpose or function and that "everything that is not necessary to that purpose or function [is] part of the expression of the idea."[28] This sort of reasoning was criticized as simplistic and not a good reflection of how a program operates.[29] Instead, courts today generally use some variation of the abstraction-filtration-comparison test, which the Second Circuit set out in 1992 in *Computer Associates International, Inc. v. Altai, Inc.*[30] In the first step of the abstraction-filtration-comparison test, the court breaks the program down into a series of functionally distinct modules. In the second step, the court filters out of each module those elements of the design that are not protected by copyright: elements dictated by "considerations of efficiency, so as to be necessarily incidental to [the function of the module]; required by factors external to the program itself; or taken from the public domain."[31] Finally, in the third step, the court compares the remaining "golden nugget" of protectable material in the original program to the allegedly infringing program.[32]

The beauty of the abstraction-filtration-comparison test lies in the fact that it mirrors the creative process of a computer programmer; the test unpacks the "largely incremental and cumulative" work that goes into designing and developing a program.[33] This fact was not lost on the

Under the *scenes a faire* doctrine, expressions that are standard to a given genre or style lack the necessary originality for copyright protection. *See* 4 *id.* § 13.03[B][4] (discussing the *scenes a faire* doctrine).

28. Whelan Assocs., Inc. v. Jaslow Dental Lab., Inc., 797 F.2d 1222, 1236 (3d Cir. 1986) (emphasis omitted).

29. Professor Nimmer, for example, asserted, "[t]he crucial flaw in this reasoning is that it assumes that only one 'idea,' in copyright law terms, underlies any computer program, and that . . . everything else must be expression." 4 NIMMER & NIMMER, *supra* note 24, § 13.03[F][1]; *see also* Sega Enter. Ltd. v. Accolade, Inc., 977 F.2d 1510, 1525 (9th Cir. 1992) ("[T]he *Whelan* rule . . . has been widely—and soundly—criticized as simplistic and overbroad.").

30. 982 F.2d 693 (2d Cir. 1992); *see also* Gates Rubber Co. v. Bando Chem. Indus., Ltd., 9 F.3d 823, 834–39 (10th Cir. 1993) (examining in detail the three steps in the abstraction-filtration-comparison test). The *Altai* court took the lead for the abstraction-filtration-comparison test from Judge Learned Hand, who advocated an "abstractions" test for distilling "patterns of increasing generality" out of a play until a point is reached "where [the patterns] are no longer protected, since otherwise the playwright could prevent the use of his 'ideas,' to which, apart from their expression, his property is never extended." Nichols v. Universal Pictures Corp., 45 F.2d 119, 121 (2d Cir. 1930).

31. *Altai*, 982 F.2d at 707.

32. *Id.* at 710.

33. Samuelson et al., *supra* note 26, at 2330. The *Altai* court noted that the analysis "resembles reverse engineering on a theoretical plane." *Altai*, 982 F.2d at 707. "[I]t is

Ninth Circuit when, less than a year after the Second Circuit decided *Altai*, the court considered a case of first impression involving reverse engineering, *Sega Enterprises Ltd. v. Accolade, Inc.*[34] Accolade programmers disassembled the object code in Sega's video game cartridges in order to discover how the games interacted with Sega's game console.[35] The programmers wrote a "development manual," incorporating the information they had discovered but leaving out any specific portions of Sega's code.[36] Next, referring to the development manual, the programmers created new games that were compatible with Sega's game console, but did not duplicate any copyrightable aspects of Sega's code.[37]

The court held that Accolade's reverse engineering of Sega's program constituted fair use, not because the "wholesale copying" of Sega's code itself satisfied the four fair use factors,[38] but because Accolade had a legitimate reason to study the unprotected ideas and functional concepts underlying Sega's code,[39] and reverse engineering

necessary essentially to retrace and map each of the designer's steps—in the opposite order in which they were taken during the program's creation." *Id.*

34. 977 F.2d 1510, 1519 (9th Cir. 1992).

35. *Id.* at 1514–15.

36. *Id.* at 1515.

37. *Id.* at 1515–16. Accolade decided not to gain information about Sega's program directly from Sega, because to do so, Accolade would have had to become a licensee of Sega. The license would have required that Sega be the exclusive manufacturer of all the interoperable games Accolade produced. *Id.* at 1514.

38. The court did review all four fair use factors, but judged the first and fourth factors—the purpose and character of the use and the effect on the plaintiff's potential market—from the perspective of Accolade's new video games instead of from the perspective of the intermediate copies Accolade made when it reverse engineered Sega's games. The court found that the first and fourth factors weighed in Accolade's favor because of the public benefit derived from allowing Accolade to market independently designed video game programs that would run on Sega's console. *Id.* at 1522–24. The court gave the most weight to the second factor, the nature of the copyrighted work. The court quoted extensively from the *Altai* opinion and concluded that many aspects of Sega's program were not copyrightable. *Id.* at 1525. The third factor admittedly weighed in Sega's favor, but the court concluded that this factor "is of very little weight." *Id.* at 1526–27.

39. The law "legitimizes" the development of compatible software for two reasons. First, the underlying purpose of copyright—"to promote the Progress of Science and useful Arts"—is best served by allowing such competitive markets to develop. U.S. CONST. art. I, § 8, cl. 8. Second, consumers derive tangible benefits from the "network effect" of widely adopted industry standards. *See* Timothy S. Teter, *Merger and the Machines: An Analysis of the Pro-Compatibility Trend in Computer Software Copyright Cases*, 45 STAN. L. REV. 1061, 1067 (1993).

> [Compatibility] [1] encourages the formation of networks, through which users can exchange files[,] . . . [2] prevents user 'lock-in' because users do not have to learn a new user interface in order to switch application programs[,] . . . [3] provides each

provided the only way for Accolade to do so.[40] In essence, Accolade performed an abstraction-filtration-comparison test on Sega's video game cartridges. Rather than filtering out uncopyrightable aspects of Sega's code, however, Accolade engineers filtered out copyrightable aspects and retained what was for them the golden nugget of *unprotected* material, which they could use with impunity to develop games that were compatible with Sega's console.

Of course, in the process of reverse engineering, Accolade made intermediate copies of Sega's programs, a prima facie violation of Sega's copyright. The court stressed that "there is no evidence in the record that Accolade sought to avoid performing its own creative work. . . . [Accolade did not] simply copy Sega's code; rather, it wrote its own procedures based on what it had learned through disassembly."[41] Nonetheless, the court also noted that the holding of the case—that making intermediate copies for the legitimate purpose of studying uncopyrightable aspects of a program constitutes fair use—"does not, of course, insulate [a defendant] from a claim of copyright infringement with respect to its finished products."[42] In other words, a reverse engineer faces two copyright hurdles: infringement due to intermediate copies and infringement in the ultimate product created. The defense of fair use only applies to the making of intermediate copies. The copyright owner of the original program might allege separately that any new, interoperable program developed as a result of reverse engineering copies protected elements of the original program. The distinction between these two different claims of wrongful conduct becomes important in considering the exemption for reverse engineering under § 1201(f) of the DMCA.

Other courts have held that reverse engineering for the purpose of accessing the uncopyrightable aspects of a computer program constitutes fair use.[43] Most recently, the Ninth Circuit affirmed its *Sega*

> user with a broader choice of application programs and hardware, and as a result of that choice, competitive prices[,] . . . [and 4] enhances competition by facilitating entry into the software industry, as entrants need not develop an entire system of their own.

Id.

40. In the words of the court, "disassembly of copyrighted object code is, as a matter of law, a fair use of the copyrighted work if such disassembly provides the only means of access to those elements of the code that are not protected by copyright and the copier has a legitimate reason for seeking such access." *Sega*, 977 F.2d. at 1518.

41. *Id.* at 1522.

42. *Id.* at 1527–28.

43. *See, e.g.*, Bateman v. Mnemonics, Inc., 79 F.3d 1532, 1540 n.18 (11th Cir. 1996)

holding in *Sony Computer Entertainment, Inc. v. Connectix Corp.*,[44] another case involving video game software. In *Sony*, the defendant, Connectix, reverse engineered Sony's basic input-output system (BIOS) software in order to create an emulator that allowed users to play Sony's games on personal computers instead of on the Sony PlayStation. Once again, the court concluded that the defendant's "intermediate copying and use of Sony's copyrighted BIOS was a fair use for the purpose of gaining access to the unprotected elements of Sony's software."[45] Significantly, because Connectix developed a new platform for players, rather than new games for players to use on Sony's platform, Connectix's stance was more openly competitive than Accolade's had been vis-à-vis Sega. The *Sega* court had entertained the argument that "video game users typically purchase more than one game. . . . [It does not] seem unlikely that a consumer particularly interested in sports might purchase both Accolade's 'Mike Ditka Power Football' and Sega's 'Joe Montana Football,' particularly if the games are, as Accolade contends, not substantially similar."[46] In contrast, the *Sony* court recognized that users were likely to "substitute" Connectix's Virtual Game Station for Sony's PlayStation console.[47] The court reasoned that Connectix's platform did not merely "supplant" the Sony PlayStation console.[48] Instead, it was "a wholly new product, notwithstanding the similarity of uses and functions between the Sony PlayStation and [Connectix's] Virtual Game Station."[49] Therefore, any loss to Sony's market for its PlayStation would be due to "legitimate" competition.[50]

There are two points to underscore from the cases that have dealt with the status of reverse engineering under the Copyright Act. First, in deciding that reverse engineering is fair use, courts do not focus on *how*

(finding the *Sega* opinion "persuasive in view of the principal purpose of copyright—the advancement of science and the arts" and noting, further, that when reverse engineering accesses original, copyrightable expression, that expression "may also be denied protection where [the defendant's] use is found to be 'fair' under 17 U.S.C. § 107"); Atari Games Corp. v. Nintendo of Am. Inc., 975 F.2d 832, 843 (Fed. Cir. 1992) ("When the nature of a work requires intermediate copying to understand the ideas and processes in a copyrighted work, that nature supports a fair use for intermediate copying.").

44. 203 F.3d 596, 602 (9th Cir. 2000).

45. *Id.* Sony did not allege that the defendant's emulator itself infringed on Sony's PlayStation, and the court assumed as much. *Id.* at 604 n.7, 606.

46. *Sega*, 977 F.2d at 1523.

47. *Sony*, 203 F.3d at 607.

48. *Id.*

49. *Id.* at 606.

50. *Id.* at 607.

a defendant reverse engineers a program, but only on the fact that the defendant *must* be able to do so. Indeed, when Sony argued that Connectix could be faulted for making more intermediate copies than necessary, the court held:

> The "necessity" we addressed in *Sega* was the necessity of the method, i.e., disassembly, not the necessity of the number of times that method was applied. . . . Even if we were inclined to supervise the engineering solutions of software companies in minute detail, and we are not, our application of the copyright law would not turn on such a distinction.[51]

The court stated further that it would not "erect an artificial hurdle in the way of the public's access to the ideas contained within copyrighted software programs."[52]

Second, even though courts dealing with reverse engineering under the Copyright Act address each of the four statutory fair use factors, the thread that runs throughout the analysis of each factor—the underlying reason for the ultimate finding of fair use—is the policy argument. The *Sony* court summed up the argument: "[T]he fair use doctrine preserves public access to the ideas and functional elements embedded in copyrighted computer software programs. This approach is consistent with the 'ultimate aim [of the Copyright Act], to stimulate artistic creativity for the general public good.'"[53] It is this balance between protection and use of computer programs that Congress intended to preserve by carving out an exemption for reverse engineering in the context of the DMCA.

III. REVERSE ENGINEERING UNDER THE DIGITAL MILLENNIUM COPYRIGHT ACT

A. Legislative History

Congress enacted the DMCA in 1998 ostensibly to implement a provision of the World Intellectual Property Organization Copyright Treaty (WCT), which requires that member states comply with the following:

51. *Id.* at 605.

52. *Id.*

53. *Id.* at 603 (quoting Sony Corp. of Am. v. Universal City Studios, Inc., 464 U.S. 417, 432 (1984)). *See generally* Pamela Samuelson & Suzanne Scotchmer, *The Law and Economics of Reverse Engineering*, 111 YALE L.J. 1575, 1583–84 (2002) (discussing how courts have "treated reverse engineering as an important factor in maintaining balance in intellectual property law").

> [P]rovide adequate legal protection and effective legal remedies against the circumvention of effective technological measures that are used by authors in connection with the exercise of their rights under this Treaty or the Berne Convention and that restrict acts, in respect of their works, which are not authorized by the authors concerned or permitted by law.[54]

But in fact, the DMCA was the culmination of several years of effort on the part of the government to figure out what copyright owners needed to deal with the internet, the new "Information Superhighway."[55] Copyright industry groups asserted that they needed better legal reinforcements to protect against unauthorized access to digital copies of their copyrighted material, which they analogized to "breaking into a locked room in order to obtain a copy of a book."[56] The WCT anti-circumvention provision was modeled after one of the earliest signals of support the U.S. government gave copyright industry groups, the so-called White Paper, which was written by the federally appointed Working Group on Intellectual Property.[57]

Released in 1995, the White Paper outlined several steps the government should take to help copyright owners, including the enactment of new digital copyright legislation that would outlaw making or distributing digital circumvention technologies.[58] Supporters and opponents of such legislation had not yet reached a compromise bill when the World Intellectual Property Organization (WIPO) held a conference in 1996. Bruce Lehman, at that time Patent Commissioner and head of the Working Group on Intellectual Property, proposed that WIPO Members sign a treaty that implemented the White Paper's recommendations.[59] Ultimately, the treaty signed (the WCT) is similar to the White Paper only in limited ways.[60] But because the treaty is not self-executing, it provided supporters of the White Paper all the more reason to urge Congress to enact legislation that would provide "adequate protection" and "effective remedies" against the

54. WIPO Copyright Treaty, *adopted* Dec. 20, 1996, 36 I.L.M. 65, 71 (1997).

55. *See generally* JESSICA LITMAN, DIGITAL COPYRIGHT 89–150 (2001) (tracing the legislative history of the DMCA).

56. H.R. REP. NO. 105-551, pt. 1, at 17 (1998).

57. INFORMATION INFRASTRUCTURE TASK FORCE, INTELLECTUAL PROPERTY AND THE NATIONAL INFORMATION INFRASTRUCTURE: THE REPORT OF THE WORKING GROUP ON INTELLECTUAL PROPERTY RIGHTS (1995) [hereinafter WHITE PAPER].

58. *Id.* at 230–34, app. at 1, 5–6.

59. LITMAN, *supra* note 55, at 128–29.

60. *Id.* at 129–30.

circumvention of digital technologies.[61]

One source of tension between supporters and opponents of implementing legislation in the United States was the question of whether the new legislation should include a fair use defense to the prohibition on circumvention. The White Paper had dealt with the issue:

> [T]he proposed legislation prohibits only those devices or products, the primary purpose or effect of which is to circumvent such systems *without authority*. That authority may be granted by the copyright owner or by limitations on the copyright owner's rights under the Copyright Act.
>
> It has been suggested that the prohibition is incompatible with fair use. First, the fair use doctrine does not require a copyright owner to allow or to facilitate unauthorized access or use of a work. . . . Second, if the circumvention device is primarily intended and used for legal purposes, such as fair use, the device would *not* violate the provision, because a device with such purposes and effects would fall under the "authorized by law" exemption.[62]

Copyright industry groups held to this line of reasoning and insisted that the implementing legislation need not include any explicit fair use defense for circumvention. Meanwhile, user interest groups—consumer electronics groups, libraries, universities, encryption researchers—urged Congress to ensure that access for fair use purposes would survive enactment of the bill.[63]

---

61. *Id.* at 130–31. Litman points out that U.S. law arguably already met the standards of the Article 11 of WCT. *See id.* at 131.

62. WHITE PAPER, *supra* note 57, at 231.

63. *See, e.g.*, *WIPO Copyright Treaties Implementation Act, and Online Copyright Liability Limitation Act: Hearing on H.R. 2381 and H.R. 2280 Before the Subcomm. on Courts and Intellectual Property of the H. Comm. on the Judiciary*, 105th Cong. 242, 245 (1997) (statement of Douglas Bennett, President, Earlham College, on behalf of the Digital Future Coalition).

> [F]air use safeguards our collective interest in the flow of information . . . .
>
> . . . [Fair use] repeatedly has been recognized by the Supreme Court as essential to the work of writers and others who creatively transmogrify the earlier works of others in the alchemy that we call "Art."
>
> . . .
>
> . . . [T]he DFC proposes that Congress amend Section 107 of the Copyright Act to make clear that fair use applies to all copyrighted works, regardless of the manner in which they are lawfully distributed or used.

Id. at 49–50 (statement of Marybeth Peters, Register of Copyright, Copyright Office of the U.S., Library of Congress) ("[M]ajor area of controversy relates to the impact of section 1201 on fair use. . . . The Copyright Office agrees that it would be extremely undesirable to end up

In the end, Congress did not include a general, fair use defense to the three anti-circumvention provisions in the DMCA. The first provision enacted makes it illegal to "circumvent a technological measure that effectively controls access to a work protected under this title."[64] The second and third provisions target devices that enable technological circumvention, thereby making it illegal to "traffic in" any device designed to circumvent either access-protection or copy-protection technology, if the device is (1) primarily designed for the purpose of circumvention, (2) has only limited commercially significant purpose or use outside circumvention, or (3) is marketed with the knowledge that it will be used for purposes of circumvention.[65]

Congress did, however, include several specific exemptions to the three anti-circumvention provisions[66] as well as a separate, temporary escape-hatch mechanism for certain classes of works identified by the Librarian of Congress in a "rulemaking proceeding" that would be conducted during the two-year period after the DMCA was enacted and again every three years thereafter.[67] Legislative history confirms that Congress crafted the exemptions and rulemaking proceedings in response to the concerns expressed over continued fair use access to digitally locked works.[68] It would be inaccurate to assume, though, that Congress intended the exemptions and rulemaking proceedings to be

with a world where fair use interests were not accommodated in an optimal manner.").

64. 17 U.S.C. § 1201(a)(1)(A) (2000).

65. § 1201(a)(2), (b)(1). Although there is no provision making it illegal to circumvent copy-protection technology, the prohibition against trafficking in tools to circumvent copy-protection technology means that only people who know how to make such tools for themselves will be able to take advantage of this "gap" in protection. *See* LITMAN, *supra* note 55, at 144.

66. In addition to the exemption for reverse engineering, § 1201(f), discussed below, Congress included exemptions for non-profit libraries, archives, and educational institutions, § 1201(d); law enforcement, intelligence, and other government activities, § 1201(e); encryption research, § 1201(g); the protection of minors, § 1201(h); the protection of personally identifying information, § 1201(i); and security testing, § 1201(j).

67. The rulemaking proceeding is laid out in § 1201(a)(1)(B)–(D). A list of the classes of works currently exempted under the rulemaking proceeding is available at http://www.copyright.gov/1201 (last visited Oct. 28, 2003).

68. *See, e.g.*, H.R. REP. NO. 105-551, pt. 2, at 36 (1998).

> Given the threat of a diminution of otherwise lawful access to works and information, the Committee on Commerce believes that a "fail-safe" mechanism is required. This mechanism would monitor developments in the marketplace for copyrighted materials, and allow the enforceability of the prohibition against the act of circumvention to be selectively waived, for limited time periods, if necessary to prevent a diminution in the availability to individual users of a particular category of copyrighted materials.

Id.

the *only* paths available for preserving fair use under the DMCA. Legislative history reveals that Congress intended the entire DMCA to embrace the balance struck in the Copyright Act between the interests of creators and users of copyrighted works.[69] In the words of Representative Bliley:

> The Committee considered it particularly important to ensure that the concept of fair use would remain firmly established in the law. Section 1201(a)(1) . . . was crafted by the Commerce Committee to protect "fair use" and other users [sic] of information now lawful under the Copyright Act.
>
> . . . .
>
> . . . Copyright law is not just about protecting information. It's just as much about affording reasonable access to it as a means of keeping our democracy healthy and doing what the Constitution says copyright law is all about: promoting "Progress in Science and the useful Arts." If this bill ceases to strike that balance, it will no longer deserve Congress' or the public's support.[70]

Thus, the congressional intent to preserve fair use under the DMCA is clear. In the case of reverse engineering, as discussed below, the exemption Congress carved out is ambiguous and crafted too narrowly to fit the realities of how and why computer programmers engage in reverse engineering. Nevertheless, courts would have the support of legislative history if they were to interpret the DMCA to allow reverse engineering for any reason that would be defensible under fair use.

B. Section 1201(f) Exemption

Section 1201(f)(1) of the DMCA provides an exemption for the reverse engineering of computer programs:

> Notwithstanding the provisions of subsection (a)(1)(A), a person who has lawfully obtained the right to use a copy of a computer program may circumvent a technological measure that effectively controls access to a particular portion of that program for the sole purpose of identifying and analyzing those elements of the program that are necessary to achieve interoperability[71] of an independently created computer program with other programs, and that have not previously been readily available to the person

69. 144 CONG. REC. H108, 7094 (daily ed. Aug. 4, 1998) (statement of Rep. Bliley).

70. *Id.*

71. Section 1201(f)(4) defines "interoperability" as "the ability of computer programs to exchange information, and of such programs mutually to use the information which has been exchanged." § 1201(f)(4).

> engaging in the circumvention, to the extent any such acts of identification and analysis do not constitute infringement under this title.[72]

Section 1201(f)(2) supplements this exemption by allowing a person—"[n]otwithstanding" the anti-trafficking provisions in § 1201(a)(2)–(b)—to develop and employ the necessary tools to reverse engineer (again, for purposes of interoperability).[73] Section 1201(f)(3) permits a person to make the information obtained under § 1201(f)(1) and the tools developed under § 1201(f)(2) available to others (solely for the purpose of enabling interoperability).[74]

Legislative history reveals that Congress intended § 1201(f) to codify the *Sega* holding:

> The objective is to ensure that the effect of current case law interpreting the Copyright Act is not changed by enactment of this legislation for certain acts of identification and analysis done in respect of computer programs. The purpose of this section is to foster competition and innovation in the computer and software industry.[75]

Nonetheless, as it stands, the § 1201(f) exemption does not go as far as *Sega* in allowing for reverse engineering. For one thing, the *Sega* court held that reverse engineering for "a legitimate reason"—not simply for the purpose of interoperability—constituted fair use.[76] Although all of the cases that have dealt with reverse engineering thus far have involved interoperability, a case could come up when the court would need to articulate a more general, "fair access" defense for reverse engineering undertaken for a legitimate reason other than interoperability. The statutory basis for such a defense and the factors a court might weigh in considering such a defense will be discussed in below in Part III.C. The remainder of the present section deals with three other ambiguities in § 1201(f) that potentially limit the application of the reverse engineering exemption.

1. Scope

Even within the narrowly drawn limits of interoperability, there are at least three ambiguities that courts must resolve to apply the reverse engineering exemption. The first ambiguity concerns the scope of

72. 17 U.S.C. § 1201(f)(1) (2000).
73. § 1201(f)(2).
74. § 1201(f)(3).
75. S. REP. NO. 105-190, at 32 (1998) (internal citation omitted).
76. Sega Enters. Ltd. v. Accolade, Inc., 977 F.2d 1510, 1518 (9th Cir. 1992).

reverse engineering. Section 1201(f)(1) limits the scope to "a particular portion" of the program, the portion that contains the "elements of the program that are [1] necessary to achieve interoperability . . . and [2] that have not previously been readily available."[77]

There are two difficulties posed by the "particular portion" requirement. For one, it is impossible for a programmer to know ahead of time which particular portion of a program to examine. Indeed, even after reverse engineering an entire program (or, more likely, a complex of programs),[78] it is not obvious which elements of the program(s) are necessary to achieve interoperability. The programmer must take into account material that did not survive translation into object code in the first place, such as source code commentary and program specifications, to determine compatibility requirements.[79] The second difficulty stems from the fact that even when a company has published portions of a program's code and/or interface requirements so that elements that are necessary to achieve interoperability are "readily available," the company may not have obviated the need to reverse engineer the program.[80] A static examination of certain elements of a program is no substitute for an analysis of the entire program as it is operating.

To resolve the ambiguity regarding the scope of a defendant's investigation, a court should read "particular portion" and "elements . . . not . . . readily available" in conjunction with the verbs that accompany these limiting words: "identifying and analyzing."[81] To identify and analyze the elements of a program that are necessary to achieve interoperability, a programmer must look at the elements in context. Even after breaking the program down into a series of discrete modules,

77. § 1201(f)(1).

78. *See* Johnson-Laird, *supra* note 12, at 345 ("The programmer merely follows a trail of logic through the software maze as it twists back and forth until a complete mental model, and thereby understanding, is achieved. The maze consists of dozens, if not hundreds, of different pieces of software, intermixed like a giant patchwork quilt.").

79. *See* Johnson-Laird, *supra* note 14, at 899 (explaining that "reverse engineering cannot tell whether a given feature is required for current or future compatibility; it can only show whether a given feature is in current use or not").

80. *See* Johnson-Laird, *supra* note 12, at 347 ("Even in those cases where companies deliberately publish detailed internal information . . . such documentation has many discrepancies; it simply fails to provide complete and accurate information about the software *as it really exists*.").

81. § 1201(f)(1) "[A] person . . . may circumvent a technological measure that effectively controls access to *a particular portion* of that program for the sole purpose of *identifying and analyzing* those *elements* of the program that are necessary to achieve interoperability . . . and that have *not* previously been *readily available*." *Id.* (emphasis added).

the programmer must observe the signals sent between the modules and then disassemble the modules—one instruction at a time, while the program is running—in order to "identify" and "analyze" the elements necessary for interoperability. Given the complexity of the task, Congress would not have intended a court to second guess the extent of investigation involved.[82]

2. Tools

A second ambiguity a court must resolve when applying the reverse engineering exemption has to do with circumvention tools. Section 1201(f)(2) permits a person "to develop and employ" the "technological means" necessary to reverse engineer for purposes of achieving interoperability.[83] Section 1201(f)(3), then, permits a person to make the tools developed under § 1201(f)(2) "available to others . . . solely for the purpose of enabling interoperability."[84] At least one court has interpreted "available to others" narrowly to conclude that § 1201(f)(3) applies only to programmers who share tools in the process of collaborative reverse engineering and does not permit "public dissemination of means of circumvention."[85]

The problem with interpreting "available to others" narrowly is that interoperable programs that are "clones" of an original program necessarily include a circumvention tool; the programmer embeds in the

82. *Cf.* Sony Computer Entm't v. Connectix Corp., 203 F.3d 596, 605 (9th Cir. 2000) (declining "to supervise the engineering solutions of software companies in minute detail").

83. Section 1201(f)(2) states in full:

> Notwithstanding the provisions of subsections (a)(2) and (b), *a person may develop and employ technological means* to circumvent a technological measure, or to circumvent protection afforded by a technological measure, in order to enable the identification and analysis under paragraph (1), or for the purpose of enabling interoperability of an independently created computer program with other programs, if such means are necessary to achieve such interoperability, to the extent that doing so does not constitute infringement under this title.

§ 1201(f)(2) (emphasis added).

84. Section 1201(f)(3) states in full:

> The information acquired through the acts permitted under paragraph (1), and *the means permitted under paragraph (2), may be made available to others* if the person referred to in paragraph (1) or (2), as the case may be, provides such information or means solely for the purpose of enabling interoperability of an independently created computer program with other programs, and to the extent that doing so does not constitute infringement under this title or violate applicable law other than this section.

§ 1201(f)(3) (emphasis added).

85. Universal City Studios, Inc. v. Reimerdes, 111 F. Supp. 2d 294, 320 (S.D.N.Y. 2000), *aff'd*, Universal City Studios, Inc. v. Corley, 273 F.3d 429 (2d Cir. 2001).

new program the instructions necessary for the new program to hook up to other programs in the same way as the original.[86] The instructions often take the form of an authentication sequence, or "secret handshake." In *Sega*, for example, Accolade embedded in its new game cartridges a verbatim authentication sequence it had discovered in Sega's games (twenty to twenty-five bytes of initialization code plus the letters "S-E-G-A"), which allowed the games to hook up to Sega's console.[87] Thus, if § 1201(f)(3) affords tools to be "made available" only to other programmers, a company like Accolade, which embeds a "secret handshake" in its new, interoperable program, faces a quandary: marketing the new program may be tantamount to "trafficking in" a tool that circumvents access, thereby violating § 1201(a)(2).[88]

To resolve the ambiguity regarding the phrase "available to others," a court should consider the relationship between § 1201(f)(2) and § 1201(f)(3). It is clear that § 1201(f)(2) allows either a third party or the programmer taking advantage of the exemption in § 1201(f)(1) to develop the tools necessary for the job. Despite the use of the singular noun in § 1201(f)(2)—"*a person* may develop and employ technological

86. There are two types of interoperability: "vertical" and "horizontal." Vertical interoperability allows a new program to communicate with the original program. Horizontal interoperability allows a new program to communicate with other programs in the same way as the original program—to be, in effect, a "clone" of the original. A single program might involve both vertical and horizontal interoperability. Accolade's new games, for example, involved vertical interoperability with Sega's console and horizontal interoperability with Sega's games. *See* Gary R. Ignatin, *Let the Hackers Hack: Allowing the Reverse Engineering of Copyrighted Computer Programs to Achieve Compatibility*, 140 U. PA. L. REV. 1999, 2042–44 (1992) (discussing vertical and horizontal interoperability).

87. Sega Enters. Ltd. v. Accolade, Inc., 977 F.2d 1510, 1516 (9th Cir. 1992). Courts generally have found such lock-and-key mechanisms to be purely functional, nonprotectable parts of a computer program. *See, e.g.*, Lexmark Int'l v. Static Control Components, 387 F.3d 522, 537–44 (6th Cir. 2004) (finding that to the extent functionality, compatibility, and efficiency requirements preclude the possibility of making any material changes to a program that functions as a lock-out code, the program is not copyrightable).

88. A district court recently granted summary judgment for the plaintiff on precisely this point. The defendant reverse engineered the plaintiff's software to create interoperable software that allowed users to play games on a website that emulated the gaming service available on the plaintiff's site. The court held that because the interoperable software (which the defendant made free and available to anyone) always sent an "okay" in response to the "CD Key" information sent by a player's game, the defendant had violated the trafficking provision of § 1201(a)(2). Davidson & Assoc., Inc. v. Internet Gateway, 334 F. Supp. 2d 1164, 1172–73, 1186 (E.D. Mo. 2004); *see also* Carla Meninsky, *Locked Out: The New Hazards of Reverse Engineering*, 21 J. MARSHALL J. COMPUTER & INFO. L. 591, 611 (2003) ("While the text of the DMCA implies that the reverse engineer may validly embed the developed key into his new product and then offer the combination for sale, if a court does focus only on the component key, this limitation could potentially expose the reverse engineer to the anti-trafficking provisions.").

means to circumvent a technological measure"—the disjunctive construction later in the sentence—"in order to enable the identification and analysis under paragraph (1), *or* for the purpose of enabling interoperability of an independently created computer program with other programs"—confirms that the person might be either helping someone else *or* engaging in reverse engineering (for the purpose of interoperability) himself.[89] The provision in § 1201(f)(3), then, which allows "the means permitted under paragraph (2) [to] be made available to others,"[90] is redundant if "available to others" means only *available to other programmers in the process of reverse engineering the program*. To be sure, § 1201(f)(3) does underscore the possibility of a collaborative effort to develop an interoperable program.[91] But to avoid complete redundancy, Congress might also have intended § 1201(f)(3) to provide that a circumvention tool, where necessary, may be embedded in a new, interoperable program that is marketed to the public. One of the rules of statutory construction is that different portions of the same statute should not be interpreted to be redundant.[92] In the context of § 1201(f), interpreting § 1201(f)(3) to allow circumvention tools to be embedded in interoperable software not only avoids overlap with § 1201(f)(2), but also makes the reverse engineering exemption correspond more closely to the *Sega* holding.

3. Character

A third ambiguity that courts must resolve to apply the § 1201(f) exemption involves the character of the interoperable program that is the product of the reverse engineering process. Section 1201(f)(1) states (and § 1201(f)(2)–(3) reiterate) that the interoperable program must be "independently created."[93] This is an oxymoron, because the new program must, of course, be dependent on the original to the extent they are interoperable. What Congress apparently intended by the phrase is that the new program must not infringe on the copyrightable aspects of the original. "Independently created" is a term of art in

89. § 1201(f)(2) (emphasis added).

90. § 1201(f)(3).

91. *See* S. REP. NO. 105-190, at 33 (1998) ("[D]eveloping complex computer programs often involves the efforts of many persons. . . . [Section 1201(f)(3)] allows developers of independently created software to rely on third parties . . . to develop the necessary circumvention tools . . . .").

92. *See* Gustafson v. Alloyd Co., Inc., 513 U.S. 561, 574 (1995) ("The Court will avoid a reading which renders some words altogether redundant.").

93. § 1201(f)(1).

copyright law, meaning that a work possesses the necessary modicum of original expression to warrant protection under the Copyright Act.[94] This interpretation of "independently created" in § 1201(f) is supported by the legislative history.[95]

The real ambiguity lies with the issue of whether a court must establish the non-infringing character of the new program before or after it applies the § 1201(f)(1) exemption. So far, courts have assumed that the question of infringement is predicate to the question of applying the reverse engineering exemption.[96] In *Sega*, however, the question of infringement was a separate inquiry following the question of whether the defense of fair use applied to reverse engineering. The court went out of its way to underscore this separate approach. After concluding that under the circumstances reverse engineering was fair use "as a matter of law,"[97] the court went on: "Our conclusion does not, of course, insulate [the defendant] from a claim of copyright infringement with respect to its finished products."[98] By keeping the analysis of the process of reverse engineering separate from the analysis of the product, the court avoided making a false inferential leap.[99] If a programmer has a legitimate reason to study someone else's program and no way to access the inner design except through reverse engineering, then, following *Sega*, the programmer's making of intermediate copies is per se legal. Nonetheless, if the same

94. *See, e.g.*, Stromback v. New Line Cinema, 384 F.3d 283, 294 (6th Cir. 2004) (discussing how a court should filter out unoriginal, unprotectable elements of a copyrighted work, "elements that were not *independently created* by the inventor," before comparing the work to an allegedly infringing work (emphasis added)).

95. *See* S. REP. NO. 105-190, at 32 (1998) ("The resulting product must be a new and original work, in that it may not infringe the original computer program.").

96. *See* Davidson & Associates, Inc. v. Internet Gateway, 334 F. Supp. 2d 1164, 1183–85 (E.D. Mo. 2004). Most recently, in *Davidson*, the court granted summary judgment for the plaintiff on the question of a § 1201(a)(1) violation, in part, because the court agreed with the plaintiff that the defendant's program was infringing and, therefore, the exemption in § 1201(f)(1) could not apply. *Id.*

97. Sega Enter. Ltd. v. Accolade, Inc., 977 F.2d 1510, 1527–28 (9th Cir. 1992). "[W]here disassembly is the only way to gain access to the ideas and functional elements embodied in a copyrighted computer program and where there is a legitimate reason for seeking such access, disassembly is a fair use of the copyrighted work, as a matter of law." *Id.*

98. *Id.* at 1528.

99. *See* Johnson-Laird, *supra* note 12, at 332.

> Both the EC and U.S. judges are making a false inferential leap that confuses the end product with the development process used to produce it. If the resulting computer software is strikingly similar to the original, then it does not matter what the process was—the program will always be infringing.

Id.

programmer goes on to make an interoperable program and a court finds, through a nuanced comparison of the two programs, that the programmer (perhaps, unintentionally) copied protectable elements of the original, then the programmer will be liable for infringement with respect to the interoperable program.

Congress might well have intended courts to follow a similar approach in applying the reverse engineering exemption in § 1201(f). In essence, Congress collapsed the underlying facts and the two elements of the *Sega* test into a single, threshold question for a court to ask: Did the defendant reverse engineer the plaintiff's program with the "purpose" of creating a competing, interoperable program?[100] If the answer is "yes," then Congress directed the court to apply the exemption "to the extent" that the defendant's program does not infringe on the plaintiff's program.[101] The words "to the extent" invite the court to make a careful, fact-intensive comparison of the two programs at issue. If Congress had intended the question of infringement to be a threshold question, it would have opened this same phrase with "as long as" or "provided that." Arguably, by using "to the extent," Congress gave a court the opportunity, through ad hoc analysis, to apply some measure of the reverse engineering exemption to an individual whose "purpose" was to create an interoperable program. To the extent the new program does not, in fact, copy protectable elements of the plaintiff's program, the defendant's reliance on self-help measures to gain access to the unprotectable elements of the plaintiff's program should be exempt from circumvention liability. To the extent the new program *does* infringe on the plaintiff's program, the protection of § 1201(f) is removed, and the defendant should be liable under both the Copyright Act and the DMCA.

100. 17 U.S.C. § 1201(f)(1) (2000).

> [A] person who has lawfully obtained the right to use a copy of a computer program may circumvent a technological measure that effectively controls access to . . . that program *for the sole purpose of identifying and analyzing those elements of the program that are necessary to achieve interoperability* . . . .

Id. (emphasis added).

101. *Id.* "[A] person . . . may circumvent . . . for the sole purpose of identifying and analyzing those elements . . . that are necessary to achieve interoperability . . . *to the extent* any such acts of identification and analysis do not constitute infringement under this title." *Id.* (emphasis added).

C. "Fair Access" Defense

1. Rationale

Although all of the reverse engineering cases that have reached courts thus far have involved a defendant who reverse engineered the plaintiff's program in order to identify the elements necessary to achieve interoperability, legal commentators agree that traditional reverse engineering practices encompass a broader range of purposes than these cases indicate.[102] A case could come up when reverse engineering would be protected under the Copyright Act by the defense of fair use, but would not be protected under the DMCA's exemption for reverse engineering.

Arguably, the DMCA leaves room for courts to develop an affirmative, "fair access" defense to deal with such a case. Even though Congress provided an express exemption for reverse engineering, it did so with the understanding that the anti-circumvention provisions, themselves, "fully respect[] and extend[] into the digital environment the bedrock principle of 'balance' in American intellectual property law for the benefit of both copyright owners and users."[103] Accordingly, courts may look to the anti-circumvention provisions to find grounds for a fair access defense for reverse engineering for purposes that fall outside the safe harbor of § 1201(f).

A similar effort to preserve the "bedrock principle" of balance between protection and use in intellectual property law lies behind the judicial development of the fair use defense for infringement under the Copyright Act. Fair use originated in eighteenth-century England in cases in which common law courts allowed for the "fair abridgment" of works of authorship without the consent of the copyright owner.[104] In

102. *See, e.g.*, Seungwoo Son, *Can Black Dot (Shrinkwrap) Licenses Override Federal Reverse Engineering Rights?: The Relationship Between Copyright, Contract, and Antitrust Laws*, 6 TUL. J. TECH. & INTELL. PROP. 63, 104–05 (2004) (pointing out that a technician may need to "break the code for many purposes other than for seeking to achieve interoperability," for instance, to correct errors or to develop highly specialized software that is not functionally interchangeable with the original program); Jeffrey D. Sullivan & Thomas M. Morrow, *Practicing Reverse Engineering in an Era of Growing Constraints Under the Digital Millennium Copyright Act and Other Provisions*, 14 ALB. L.J. SCI. & TECH. 1, 4–5 (2003) ("Recent legislation aimed at protecting digital rights holders has been drawn extremely—and perhaps unintentionally—broadly, and may call into question the lawfulness of numerous reverse engineering practices that would previously have struck most technical and legal observers as routine and beyond sanction.").

103. H.R. REP. NO. 105-551, pt. 2, at 26 (1998); *see* discussion *supra* Part III.A.

104. *See* Lydia Pallas Loren, *Redefining the Market Failure Approach to Fair Use in an*

the United States, the first reported case to involve fair use was *Folsom v. Marsh*,[105] in which Justice Story summarized the inquiry:

> In short, we must often, in deciding questions of this sort, look to the nature and objects of the selections made, the quantity and value of the materials used, and the degree in which the use may prejudice the sale, or diminish the profits, or supersede the objects, of the original work.[106]

Although fair use went on to become, as one court famously called it, the "most troublesome [doctrine] in the whole law of copyright,"[107] courts have never questioned the necessity of the defense. Indeed, every time Congress has revised the Copyright Act and expanded the nature of the protection afforded copyright owners, courts have found all the more reasons to advance fair use,[108] because it "permits courts to avoid rigid application of the copyright statute when, on occasion, it would stifle the very creativity which that law is designed to foster."[109]

Congress signaled its approval of the fair use doctrine when it codified the defense in § 107 of the Copyright Act of 1976.[110] Yet it did so cautiously, intending that the doctrine should not be "frozen" in the Act, but should remain in the hands of the judiciary, so that it could

---

Era of Copyright Permission Systems, 5 J. INTELL. PROP. L. 1, 13–22 (tracing the origins of the fair use doctrine).

105. Folsom v. Marsh, 9 F. Cas. 342, 348 (C.C.D. Mass. 1841). Justice Story used the phrase "fair abridgment," not "fair use." Nine reported cases later in the nineteenth century do use the phrase "fair use." Laura G. Lape, *Transforming Fair Use: The Productive Use Factor in Fair Use Doctrine*, 58 ALB. L. REV. 677, 680 (1995).

106. 9 F. Cas. at 348.

107. Deller v. Samuel Goldwyn, Inc., 104 F.2d 661, 662 (2d Cir. 1939).

108. Congress enacted the first Copyright Act in 1790. Congress revised the Act in 1802, 1831, 1870, 1909, and 1976. *See* Loren, *supra* note 104, at 17–18 ("As more rights were added to the rights granted to copyright owners, fair use was asserted more frequently . . . .").

109. Iowa State Univ. Research Found., Inc. v. Am. Broad. Co., 621 F.2d 57, 60 (2d Cir. 1980).

110. 17 U.S.C. § 107 (2000). We see the far-reaching influence of Justice Story's formulation of the fair use inquiry in the list of factors Congress included in § 107:

> In determining whether the use made of a work in any particular case is a fair use the factors to be considered shall include—
>
> (1) the purpose and character of the use . . . ;
>
> (2) the nature of the copyrighted work;
>
> (3) the amount and substantiality of the portion used in relation to the copyrighted work as a whole; and
>
> (4) the effect of the use upon the potential market for or value of the copyrighted work.

§ 107; *see also* Pierre N. Leval, Commentary, *Toward a Fair Use Standard*, 103 HARV. L. REV. 1105 (1990) (comparing the statutory factors in § 107 to Justice Story's formulation of the fair use inquiry).

more easily respond to technological changes.[111] When Congress enacted the DMCA, some wondered whether § 1201(c)(1), which provides that "[n]othing in this section shall affect rights, remedies, limitations, or defenses to copyright infringement, *including fair use*, under this title,"[112] signaled Congress's intent that courts should extend the fair use defense into the realm of digital circumvention.[113] However, a close reading of § 1201(c)(1) does not bear out this interpretation. Circumvention is distinct from copyright infringement. Section 1201(c)(1) only applies following lawful access to a copyrighted digital work, not as a defense for obtaining unauthorized access to a work.[114] On the other hand, § 1201(c)(1) does underscore Congress's continued commitment to the judicial development of the fair use doctrine. This commitment, itself, lends support to a parallel judicial development of a fair access defense.[115]

2. Precedent

The one court to consider the possibility of applying the defense of fair use to a DMCA violation unequivocally decided that the defense did not apply: "Technological access control measures have the capacity to prevent fair uses of copyrighted works as well as foul."[116] However, that case involved a bad actor, a defendant who was "viewed as a leader of the computer hacker community" and who, in an act of "electronic civil disobedience," supported links to websites that offered

111. *See* H.R. REP. NO. 94-1476, at 67 (1976).

> The bill endorses the purpose and general scope of the judicial doctrine of fair use, but there is no disposition to freeze the doctrine in the statute, especially during a period of rapid technological change. [T]he courts must be free to adapt the doctrine to particular situations on a case-by-case basis. Section 107 is intended to restate the present judicial doctrine of fair use, not to change, narrow, or enlarge it in any way

Id.

112. 17 U.S.C. § 1201(c)(1) (2000) (emphasis added).

113. *See* Universal City Studios, Inc. v. Corley, 273 F.3d 429, 443 (2d Cir. 2001) ("We disagree that subsection 1201(c)(1) permits such a reading. Instead, it simply clarifies that the DMCA targets the *circumvention* of digital walls guarding copyrighted material . . . but does not concern itself with the *use* of those materials after circumvention has occurred.").

114. *See* 4 NIMMER & NIMMER, *supra* note 24, § 13.05[F][6] ("Section 1201 . . . defines the anti-circumvention as something distinct from copyright infringement. For that reason, fair use is no defense to an action brought under Section 1201.").

115. *Cf.* 4 *id.* § 13.05[D][4] n.513.13 (suggesting that when case law under the fair use doctrine diverges from interpretations of the DMCA, the DMCA might be construed expansively to cover non-exploitative uses of reverse engineering tools).

116. Universal City Studios, Inc. v. Reimerdes, 111 F. Supp. 2d 294, 304 (S.D.N.Y. 2000), *aff'd*, Universal City Studios, Inc. v. Corley, 273 F.3d 429 (2d Cir. 2001).

DeCSS, a circumvention tool, for download.[117] Furthermore, the defendant did not argue fair use for himself, but for "those who wish to gain access to technologically protected copyrighted works in order to make fair—that is, non-infringing—use of them."[118] Thus, the facts of the case are not on point with a hypothetical case in which the defendant has reverse engineered a computer program for non-interoperability, but fair-use-defensible reasons.

Two other recent court of appeals cases offer more helpful precedent for a fair access defense for reverse engineering.[119] Both cases involved defendants who bypassed the digital lock-and-key mechanism in software that the plaintiffs had embedded in consumer goods in order to market competing replacement parts for the goods. In both cases, the courts ruled that the DMCA could not be used to prevent a legitimate use of an otherwise accessible copyrighted work. And in both cases, the courts did not base their reasoning on the fair use defense, but on a close reading of the anti-circumvention provisions themselves, particularly the words "access," "protection," and "authority" in § 1201(a).

In the first case, the plaintiff, Chamberlain, manufactured garage door openers (GDOs) and used "rolling-code" software within the handheld transmitters it sold with the GDOs ostensibly to prevent criminals from "grabbing" the code and opening people's garages.[120] The defendant, Skylink, marketed a universal GDO transmitter that bypassed rolling-code software.[121] Because rolling-code technology is based on a process that is relatively straightforward and familiar to computer programmers, Skylink was able to develop its method for bypassing the rolling-code software in Chamberlain's transmitters without actually reverse engineering the software.[122] Consequently, Chamberlain did not allege that Skylink had either infringed its

117. *Id.* at 308, 312. DeCSS is a computer program that circumvents the CSS protection system on motion pictures distributed on DVDs. The defendant in this case had not developed DeCSS himself; thus, DeCSS could not take advantage of § 1201(f)(3). The court concluded that § 1201(f)(3) "permits information acquired through reverse engineering to be made available to others only by the person who acquired the information." *Id.* at 320.

118. *Id.* at 304.

119. Chamberlain Group, Inc. v. Skylink Techs., Inc., 381 F.3d 1178 (Fed. Cir. 2004); Lexmark Int'l, Inc. v. Static Control Components, Inc., 387 F.3d 522 (6th Cir. 2004).

120. *Chamberlain*, 381 F.3d at 1183–85.

121. *Id.* at 1184–86.

122. Skylink Technologies, Inc.'s Opposition to the Chamberlain Group, Inc.'s Motion for Summary Judgment at 6, 8, Chamberlain Group, Inc. v. Skylink Technologies, Inc., 292 F. Supp. 2d 1023 (N.D. Ill. 2003) (No. 02-C-6376), 2003 WL 22961236.

copyright or circumvented its technology, but only that Skylink had violated the DMCA's anti-trafficking provision.[123] In other words, according to Chamberlain, Skylink was trafficking in a device that enabled homeowners to circumvent a technological measure (the rolling-code software) that effectively controlled access to the copyrighted computer program within Chamberlain's GDOs.

The Federal Circuit affirmed the lower court's grant of summary judgment in favor of Skylink, holding that Chamberlain had failed to show that Skylink's universal GDO transmitter enabled anything more than a homeowner's legitimate use of the copyrighted software embedded in Chamberlain's GDOs.[124] The court observed that in marketing its GDOs, Chamberlain placed no restrictions on a homeowner's ability to use replacement transmitters. Indeed, to do so would be to go against market norms, because homeowners have "long been able" to use universal transmitters with their GDO systems.[125]

Lack of notice, however, was not the reason Chamberlain failed to persuade the court of the merits of its claim. Instead, the court said that Chamberlain had fundamentally misconstrued Congress's intent in enacting the DMCA. The court focused on the language of § 1201(a)(2)(A), specifically on what it means to "circumvent[] a technological measure that effectively controls *access* to a work *protected* under this title."[126] The court explained that the "access" with which the DMCA is concerned is intimately bound up with the rights that the Copyright Act grants to copyright owners, that is, rights that are "protected" under the Copyright Act and not rights that the Copyright Act grants to the public.[127] Thus, the court declared that a copyright owner could not "block *all* access" to its copyrighted work, but only access that would enable an infringing use of the work.[128]

As a corollary to its analysis of the connection between "access" and "protection" in § 1201(a)(2)(A), the court discussed the notion of authorization, which it said "is central to understanding § 1201(a)."[129]

123. *See* 17 U.S.C. § 1201(a)(2) (2000). Section 1201(a)(2) prohibits a person from trafficking in a device that (1) is primarily designed to circumvent a technological measure that effectively controls access to a copyrighted work, (2) has only limited commercially significant purpose or use outside circumvention, and (3) is marketed with the knowledge that it will be used for purposes of circumvention. *Id.*

124. *Chamberlain*, 381 F.3d at 1202–04.

125. *Id.* at 1183.

126. § 1201(a)(2)(A) (emphasis added).

127. *Chamberlain*, 381 F.3d at 1197–1204.

128. *Id.* at 1199.

129. *Id.* at 1202.

Noting that the DMCA "*defines* circumvention as an activity undertaken 'without the authority of the copyright owner,'"[130] the court reasoned that in some instances, the requisite authority lies in the Copyright Act.[131] "Copyright law itself authorizes the public to make certain uses of copyrighted materials. Consumers who purchase a product containing a copy of embedded software have the inherent legal right to use that copy of the software. What the law authorizes, Chamberlain cannot revoke."[132]

The second recent circuit court case that provides precedent for a fair access defense also involved aftermarket replacement parts. The plaintiff, Lexmark, manufactured printer cartridges that contained a microchip designed to prevent its printers from functioning with cartridges it had not refilled.[133] The defendant, Static Control Components (SCC), designed a microchip that mimicked Lexmark's microchip and then marketed the chip to companies that sold remanufactured printer cartridges.[134] Lexmark alleged that SCC's chip violated § 1201(a)(1)(A) of the DMCA,[135] because it circumvented a technological measure—an authentication sequence or "secret handshake"—that controlled access both to the Toner Loading Program (TLP) located on Lexmark's microchip and to the Printer Engine Program (PEP) located within Lexmark's printers.[136]

In reversing the lower court's entry of a preliminary injunction against SCC, the Sixth Circuit found that Lexmark's TLP probably did not satisfy the originality requirement of the Copyright Act (the case was remanded on this point).[137] If the TLP were not copyrightable,

130. *Id*. at 1193 (quoting § 1201(a)(3)(A)).

131. *Id.* at 1193–94. The court's reasoning follows the reasoning set out in the White Paper: "[I]f the circumvention device is primarily intended and used for legal purposes, such as fair use, the device would *not* violate the provision, because a device with such purposes and effects would fall under the 'authorized by law' exemption." WHITE PAPER, *supra* note 57, at 231.

132. *Chamberlain*, 381 F.3d at 1202.

133. Lexmark Int'l, Inc. v. Static Control Components, Inc., 387 F.3d 522, 529 (6th Cir. 2004).

134. *Id.*

135. § 1201(a)(1)(A). "No person shall circumvent a technological measure that effectively controls access to a work protected under this title." *Id.*

136. SCC did not reverse engineer Lexmark's microchip, but "slavishly copied" the Toner Loading Program located on the chip and replaced the authentication sequence that was also located on the chip with a different, publicly available encryption program that enabled interoperability of its chip with Lexmark's printers. *Lexmark*, 387 F.3d at 530–31, 538.

137. *Id.* at 537–44.

there could be no DMCA violation of the TLP because the anti-circumvention provisions only apply to works "protected under this title."[138] Even if the TLP were copyrightable, though, the court held that the authentication sequence located on Lexmark's microchip did not control "access" either to the TLP or to the copyrightable PEP within Lexmark's printers.

Like the Federal Circuit in *Chamberlain*, the Sixth Circuit focused on what it means to "circumvent a technological measure that effectively controls *access* to a work protected under this title."[139] The court noted that the dictionary definition of "access" was "'the ability to enter, to obtain, or to make use of,'" and reasoned that the relevant meaning of "access" in the case of a computer program was "'the ability to . . . obtain' a copy of the work or to 'make use of' the literal elements of the program (its code)."[140] Lexmark argued that the authentication sequence on its microchip controlled access to its programs, because the sequence controlled a consumer's ability to "make use of" the programs.[141] The court disagreed. In the court's view, it was the purchase of a Lexmark printer (and printer cartridge) that controlled "access" to the programs.[142] The court pointed out:

> Anyone who buys a Lexmark printer may read the literal code of the Printer Engine Program directly from the printer memory, with or without the benefit of the authentication sequence, and the data from the program may be translated into readable source code after which copies may be freely distributed. . . . No security device, in other words, protects access to the Printer Engine Program Code and no security device accordingly must be circumvented to obtain access to that program code.[143]

The line of reasoning in these two cases creates precedent for the argument that a person who lawfully obtains a copy of a computer program has the "authority" granted under the Copyright Act's fair use doctrine to "access" or "make use of" the program's code. In theory, this authority places fair-use-defensible reverse engineering outside the definition of circumvention and, hence, outside the reach of the DMCA. That is, the act of reverse engineering, if done for legitimate purposes, is

138. § 1201(a)(1)(A).

139. *Id.* (emphasis added).

140. *Lexmark*, 387 F.3d at 546 (quoting MERRIAM-WEBSTER'S COLLEGIATE DICTIONARY 6 (10th ed. 1999)). The DMCA does not define "access."

141. *Lexmark*, 387 F.3d at 546.

142. *Id.*

143. *Id.* at 546–47.

done under the authority of the copyright owner (via fair use) and therefore does not violate § 1201(a)(1)(A).

In practice, a defendant is more likely to find success with the fair access argument if it is couched as an affirmative defense, rather than as a failure to state a claim. Reverse engineering is more complicated than opening one's garage door or replacing the toner cartridge in one's printer, and a court is likely to assume that reverse engineering involves, by its very nature, the circumvention of a technological measure. Raising fair access as an affirmative defense also has the advantage of drawing a court's attention to the parallels between fair access and fair use. So, for example, cases that rely on the "reasonable and customary" theory of fair use lend support to the argument that reverse engineering that does not involve "transformative" access to the original program (i.e., the creation of a new, interoperable program) might nonetheless constitute fair access if undertaken for purposes that are widely accepted or socially beneficial.[144]

3. Factors

Just how far the fair access defense for reverse engineering might go is something courts will need to work out on a case-by-case basis. A court considering the defense should explore all the evidence at hand in light of the purposes of the DMCA: to protect copyright owners against the threat of digital piracy and, at the same time, to preserve the "bedrock principle" of balance between protection and use in intellectual property law.[145] The following discussion suggests three factors a court should address. Of these, the first—whether the reverse engineering in question led to what traditionally would be considered a fair use of the original program—carries the most weight. Labeling the use "fair," however, should not end the court's analysis. A second factor to consider is whether an inherent limitation in the market led to the defendant's need to reverse engineer the plaintiff's program.[146] This

144. The phrase "reasonable and customary" describes the use of a copyrighted work that, although technically infringing, is nevertheless fair under the theory of the author's implied consent to use of the work for a socially beneficial or widely accepted purpose. *See, e.g.*, Harper & Row Publishers, Inc. v. Nation Enters., 471 U.S. 539, 550–51 (1985); Am. Geophysical Union v. Texaco Inc., 802 F. Supp. 1, 11–13 (S.D.N.Y. 1992). The "reasonable and customary" theory of fair use lies at the opposite end of the spectrum from the "transformative" use theory, under which the use of a copyrighted work that is infringing is nevertheless justified because it adds new purpose or character to the original work. *See, e.g.*, Campbell v. Acuff-Rose Music, Inc., 510 U.S. 569, 578–79 (1994).

145. H.R. REP. NO. 105-551, pt. 2, at 26 (1998); *see* discussion *supra* Parts III.A, III.C.1.

146. *See* Wendy J. Gordon, *Excuse and Justification in the Law of Fair Use:*

factor recognizes that in every case that makes its way to court, the plaintiff will hypothetically, at least, desire to control access to the inner workings of the program in question. The court must assess the likelihood that the plaintiff would, indeed, have been able to provide access in a way that would serve the defendant's fair-use-authorized interest in observing how the program operates. Finally, the third factor, the nature of the plaintiff's program, calls for the court to evaluate the level of copyright protection the program merits and, in turn, the level of protection the program should receive under the DMCA. Just as a program that consists primarily of functional or unoriginal elements—elements that would be filtered out under the abstractions-filtration-comparison test—would receive "relatively weak" protection under the Copyright Act,[147] so the same program should arguably receive relatively weak protection under the DMCA.[148]

The primary factor a court must address under the fair access defense is whether the reverse engineering in question led to what traditionally would be considered a fair use of the plaintiff's program. Fixing bugs and reverse engineering for research purposes would qualify; reverse engineering to extract and duplicate copyrightable code in a new program would not. The court should allow expert witnesses to testify at this early stage of the case, first, to gain sufficient background knowledge to appreciate the defendant's purpose in reverse engineering and, second, to grasp some of the subtle differences between programs that may be drawn out by the abstraction-filtration-comparison test. Even if the defendant did not create a new, allegedly infringing program, the court will need to understand the inner workings of the plaintiff's program in order to analyze the four fair use factors.[149] If the court finds that the defendant's use of the plaintiff's program was, in fact, fair, the time spent carefully evaluating the fair use factors will pay

Transaction Costs Have Always Been Part of the Story, 50 J. COPYRIGHT SOC'Y U.S.A. 149, 156–60 (2003) (discussing "inherent limitations" in the market where non-economic values are at stake and market transactions could never achieve the desired goals, so the law "justifies" a defendant's proceeding without consent or compensation).

147. *See* discussion *supra* Part II; *see also* Computer Assoc. Int'l, Inc. v. Altai, Inc., 982 F.2d 693, 712 (2d Cir. 1992) ("To be frank, the exact contours of copyright protection for non-literal program structure are not completely clear. . . . Indeed, it may well be that the Copyright Act serves as a relatively weak barrier against public access to the theoretical interstices behind a program's source and object codes.").

148. The *Lexmark* court did not stop at "relatively weak" protection, but stated flatly that if the copyrightable expression of a program "operates on only one plane: in the literal elements of the program, its source and object code," then access to the program "is not covered" by the DMCA. *Lexmark*, 387 F.3d at 548.

149. *See supra* notes 24, 110.

off as the case unfolds and the information gleaned helps the court further evaluate the fair access defense.

A second factor to address under the fair access defense is whether an inherent limitation in the market led to the defendant's need to reverse engineer the program. In some cases, the interests of the parties may have been so opposed at the time the defendant reverse engineered the program, it will be evident that, absent a court order, the plaintiff would not have willingly provided the defendant access to the inner workings of the program. Consider, for example, companies that develop and market competing software. If one company suspects that the other has infringed its copyright, in order to meet the "reasonable inquiry" burden of Rule 11 of the *Federal Rules of Civil Procedure*, the company must reverse engineer the other's software before filing a lawsuit.[150] In other cases, the parties' interests may not have been opposed, but the defendant arguably would have been unable to obtain the necessary information from the plaintiff in a reasonable, timely manner. A defendant's reliance on self-help to gain access to the inner workings of the plaintiff's program to correct an error, for example, may be justifiable if the plaintiff does not offer ongoing technical support for purchasers/licensees of its program. If the plaintiff maintains that, given the opportunity, it would have helped the defendant correct the error, the court should weigh other facts and circumstances, such as the structure of the industry involved and the nature of any other contacts between the parties, to assess the likelihood that the plaintiff would have been able to meet the defendant's needs.

A third factor to address under the fair access defense is the nature of the plaintiff's program. This factor overlaps, to an extent, the inquiry under the second fair use factor.[151] The more original expression a program contains, the more copyright protection it merits. Programs that are dictated by practical realities are protected, if at all, only against verbatim copying.[152] In the context of the fair access defense, a court should go beyond measuring the level of copyright protection the plaintiff's program merits to ask what will be the consequences of denying access to the inner workings of the program. If allowing

150. See Sullivan & Morrow, *supra* note 102, for a discussion of the tension between the DMCA and Rule 11 of the *Federal Rules of Civil Procedure*.

151. 17 U.S.C. § 107 (2000). "In determining whether the use made of a work in any particular case is a fair use the factors to be considered shall include . . . (2) the nature of the copyrighted work; . . . ." *Id.*

152. *See Lexmark*, 387 F.3d at 533–37 (discussing the level of copyright protection a computer program may merit).

unrestricted public access to the copyrighted expression within a program would materially affect the plaintiff's ability or incentive to build upon that expression, such as creating follow-up programs, then the court should weigh this factor in the plaintiff's favor. On the other hand, if unrestricted access to the inner workings of a program would have little or no effect on the plaintiff's ability or incentive to build upon the copyrighted expression within the program, then the court should weigh this factor in the defendant's favor. The "deadweight loss" to society that already occurs under copyright law will be exacerbated if the DMCA is interpreted in a way that is over-inclusive, affording absolute anti-circumvention protection for programs that the Copyright Act would protect only against verbatim copying.[153]

CONCLUSION

Congress enacted the DMCA with a relatively precise idea of the protection it wanted to afford copyright owners against the threat of digital piracy and a relatively imprecise idea of how to preserve the public's fair use of digitally locked works. The exemptions and ongoing rulemaking procedures set out in the DMCA represent but two ways that Congress attempted to compensate for fair use within the strictures of the anti-circumvention provisions. Congress also left room for the courts to play an active role, first, by applying the exemptions in ways that are consistent with Congress's express intent in enacting them and, second, by interpreting the anti-circumvention provisions in ways that respect the underlying goal of the Copyright Act: to balance protection and use of copyrighted works so as "[t]o promote the Progress of Science and useful Arts."[154]

Legislative history shows that Congress intended the exemption in § 1201(f) of the DMCA to codify the holding of *Sega*, a case that involved a defendant who reverse engineered the plaintiff's program in order to create a competing, interoperable program.[155] The court in *Sega* held that when a defendant has no other way to examine the uncopyrightable aspects of a program and a legitimate reason to study them, then reverse engineering is fair use "as a matter of law."[156] The

153. *See* Wendy J. Gordon, *Authors, Publishers, and Public Goods: Trading Gold for Dross*, 36 LOY. L.A. L. REV. 159, 195 (discussing deadweight loss in the context of copyright term extension).

154. U.S. CONST. art. I, § 8, cl. 8.

155. S. REP. NO. 105-190, at 31 (1998).

156. Sega Enters. Ltd. v. Accolade, Inc., 977 F.2d 1510, 1518 (9th Cir. 1992).

Sega court reserved as a separate question whether the defendant's new program, in fact, infringed on the plaintiff's.[157] Congress reduced the *Sega* holding to a single, threshold question for a court to ask when applying the exemption in § 1201(f): Did the defendant reverse engineer the plaintiff's program with the "purpose" of creating a competing, interoperable program?[158] If the answer is "yes," then Congress directed the court to apply the exemption "to the extent" that the defendant's program does not infringe on the plaintiff's program.[159] A court should not second guess the scope of the defendant's reverse engineering, because reverse engineering is by nature an open-ended, exploratory process. Nor should the court hold the defendant liable for violating the anti-trafficking provision of the DMCA if the defendant's new program has embedded in it a device for enabling the new program to communicate with programs that normally communicate with the plaintiff's, because this type of interaction is the essence of "interoperability."

If the defendant did not reverse engineer the plaintiff's program with the purpose of creating an interoperable program, then arguably the court should go on to ask if the defendant's self-help access to the inner workings of the plaintiff's program is nonetheless justifiable under an affirmative defense of fair access. Recent case law suggests that this defense might be grounded in an interpretation of the words "access," "protection," and "authority" in § 1201(a) to mean that when a defendant has lawfully obtained a copy of the plaintiff's program, the defendant has the authority granted under the Copyright Act to make use of the program in any way that would be justified under the fair use defense. This Comment has suggested three factors a court should consider in applying the fair access defense: (1) whether the defendant's use of the plaintiff's program traditionally would be considered fair use; (2) whether an inherent limitation in the market led to the defendant's need to access the inner workings of the program without the plaintiff's permission; and (3) whether the nature of the plaintiff's program is such that it deserves relatively weak protection under the DMCA. Although this sort of fact-intensive inquiry might complicate the court's task, it is necessary to carry out Congress's express intent to preserve the fair use of digitally locked works.[160] It is also the only way for a court to reach a

157. *Id.* at 1528.

158. 17 U.S.C. § 1201(f)(1) (2000).

159. *Id.*

160. *See* Chamberlain Group, Inc. v. Skylink Techs., Inc., 381 F.3d 1178, 1202–03 (Fed. Cir. 2004).

decision that is true to the realities of how and why programmers reverse engineer.

DONNA L. LEE*

> We conclude that 17 U.S.C. § 1201 prohibits only forms of access that bear a reasonable relationship to the protections that the Copyright Act otherwise affords copyright owners. While such a rule of reason may create some uncertainty and consume some judicial resources, it is the only meaningful reading of the statute. Congress attempted to balance the legitimate interests of copyright owners with those of consumers of copyrighted products. The courts must adhere to the language that Congress enacted to determine how it attempted to achieve that balance.

Id. (citations omitted).

* B.S. 1985, Southern Adventist University; Ph.D. 1992, Duke University; J.D. Candidate 2006, Lewis & Clark Law School. The author would like to thank Professor Lydia Loren for her assistance in writing this Comment.